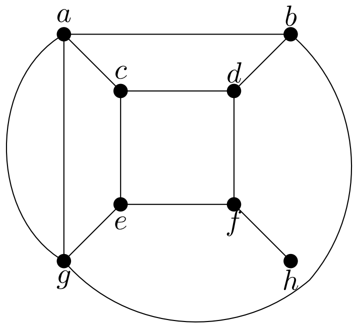
Práctica 7 - Árboles (parte 2)

1. Considere el siguiente grafo



a) Use el algoritmo de búsqueda a lo ancho (BFS) para encontrar un árbol de expansión con el orden de vértices dado en cada caso.

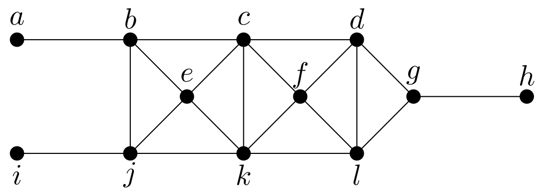
I. $hgfedcba$ II. $hfdbgeca$ III. $chbgadfe$

b) Use el algoritmo de búsqueda en profundidad (DFS) para encontrar un árbol de expansión con el orden de vértices dado en cada caso.

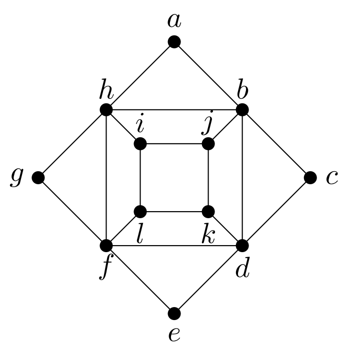
I. $hgfedcba$ II. $hfdbgeca$ III. $dhcbefag$

2. Encuentre un árbol de expansión para cada uno de los siguientes grafos.

a)



b)



3. Determine si cada una de las siguiente afirmaciones es verdadera o falsa, justificando adecuadamente.

a) Si G es un grafo conexo y T un árbol de expansión para G, entonces existe un orden de los vértices de G tal que el algoritmo BFS tiene como salida al árbol de expansión T.

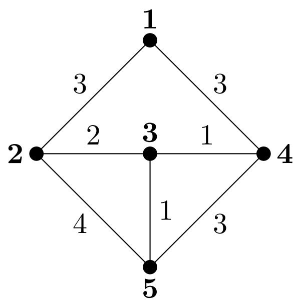
b) Si G es un grafo conexo y T un árbol de expansión para G, entonces existe un orden de los vértices de G tal que el algoritmo DFS tiene como salida al árbol de expansión T.

4. a) Escriba un algoritmo basado en el BFS para determinar si un grafo es o no conexo.

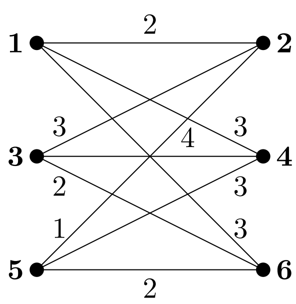
b) Escriba un algoritmo basado en el DFS para determinar si un grafo es o no conexo.

5. Encuentre un árbol de expansión mínima para cada uno de los siguientes grafos, utilizando el algoritmo de Kruskal.

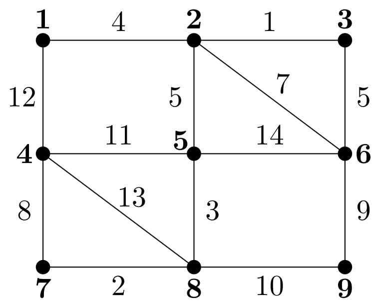
a)



b)



c)



6. a) Sea G un grafo conexo ponderado, sea v un vértice en G y sea e la arista con peso mínimo incidente en v. Demuestre que e está contenida en algún árbol de expansión mínima.

   b) Sea G un grafo conexo ponderado y sea v un vértice en G. Suponga que los pesos de las aristas incidentes en v son distintos. Sea e la arista con peso mínimo incidente en v. ¿Debe estar e contenida en todo árbol de expansión mínima?

7. Determine si cada una de las siguiente afirmaciones es verdadera o falsa, justificando adecuadamente.

   a) Si todos los pesos en un grafo conexo G son diferentes, entonces G admite un único árbol de expansión mínima.

   b) Si todos los pesos en un grafo conexo G son diferentes, entonces los árboles de expansión de G distintos tienen pesos distintos.

8. a) Sea G un grafo ponderado. Demuestre que si, mientras sea posible, se elimina una arista de G con peso máximo y cuya eliminación no desconecta a G, el resultado es un árbol de expansión mínima para G.

   b) Escriba un algoritmo que encuentre un árbol de expansión mínima en un grafo conexo ponderado utilizando el resultado del ítem anterior.

9. *Algoritmo de Prim.*

   Entrada. Un grafo conexo ponderado.

   Idea. Mantener un subgrafo conexo H e ir agregando nuevos vértices incidentes en aristas de peso mínimo.

   Inicio. Elegir $v \in V(G)$. $V(H) = \{v\}$, $E(G) = \varnothing$.

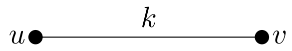
   Iteración. Mientras $V(H) \neq V(G)$, elegir la arista e de menor peso entre el conjunto de aristas que tienen un extremo en $V(H)$ y otro en $V(G)\backslash V(H)$. Si la arista elegida es $e = uw$ con $u \in V(H)$ y $w \in V(G)\backslash V(H)$, actualizamos $V(H) \leftarrow V(H) \cup \{w\}$, $E(H) \leftarrow E(H) \cup \{e\}$.

   Pruebe que si G es un grafo conexo ponderado, el algoritmo de Prim produce un árbol recubridor de G de peso mínimo.

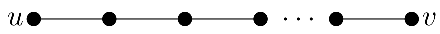
10. Encuentre un árbol de expansión mínima para cada uno de los grafos del ejercicio 5, utilizando el algoritmo de Prim.

11. a) Escriba un algoritmo basado en el BFS que encuentre la distancia, en un grafo no ponderado, desde un vértice fijo v a todos los demás.

b) Sea G un grafo ponderado en el que el peso de cada arista es un entero positivo. Sea G' el grafo obtenido a partir de G al sustituir cada arista



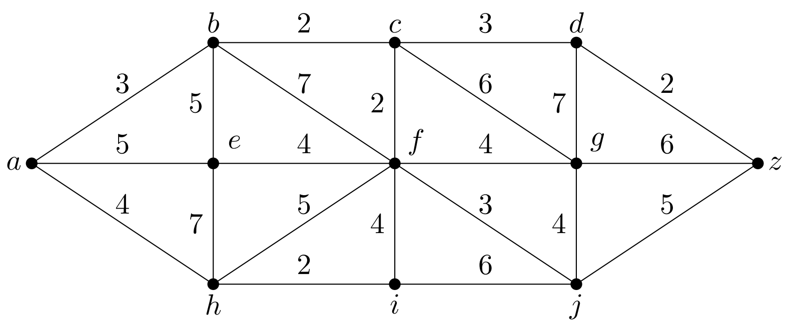
en G de peso k por un camino P_k de aristas no ponderadas



k aristas

Demuestre que el algoritmo de Dijkstra para encontrar la longitud mínima de una trayectoria en el grafo ponderado G desde un vértice fijo v a todos los demás y realizar la búsqueda a lo ancho en el grafo no ponderado G' comenzando en el vértice v son el mismo proceso.

12. Aplique el algoritmo de Dijkstra al grafo ponderado $G = (V, E)$ de la figura y determine la longitud de una ruta más corta entre los pares de vértices indicados.

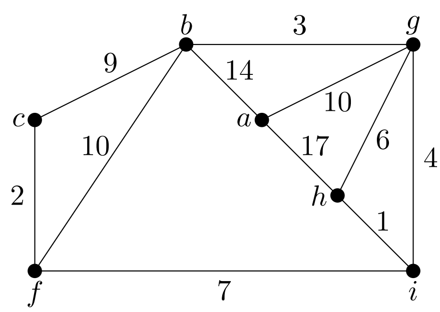


a) a y f. b) a y g. c) a y z. d) b y j. e) h y b. f) h y d.

13. a) Escriba un algoritmo que dado un grafo (ponderado) G y un vértice $v \in V(G)$, calcule $d(v,z)$ y dé un (v,z)-camino de longitud mínima para todo $z \in V(G)$.

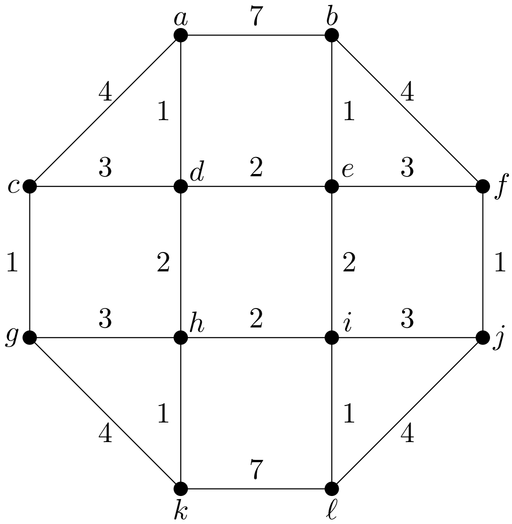
b) Escriba un algoritmo que dado un grafo (ponderado) G y dos vértices $v, w \in V(G)$, calcule $d(v,w)$.

14. a) Aplique el algoritmo de Dijkstra al grafo ponderado $G = (V, E)$ de la figura y determine la longitud de una ruta más corta del vértice a a cada uno de los otros vértices de G.



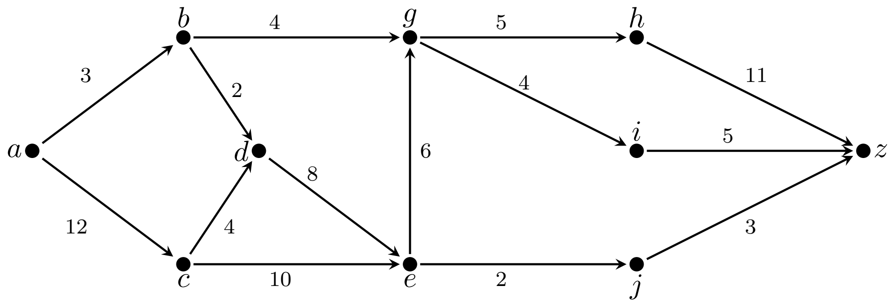
b) Determine un camino de longitud mínima del vértice a a los vértices c, f e i.

15. a) Aplique el algoritmo de Dijkstra al grafo ponderado $G = (V, E)$ de la figura y determine la longitud de una ruta más corta del vértice a a cada uno de los otros vértices de G.



b) Determine un camino de longitud mínima del vértice a a los vértices f, g y ℓ.

16. Considere el siguiente digrafo ponderado G.



Determine la longitud de una ruta más corta del vértice a a cada uno de los otros vértices de G.

17. Suponga que el algoritmo de Dijkstra tiene como entrada un grafo ponderado G que no es conexo, y un vértice inicial u. Si v es un vértice de G que no está en la misma componente conexa que u ¿qué valor tiene $t(v)$ al finalizar la ejecución del algoritmo?

18. Decida si las siguientes afirmaciones son **verdaderas** o **falsas**, justificando adecuadamente.

a) Sea $G = (V, E)$ un grafo ponderado con $V = \{v_1, v_2, \ldots, v_n\}$ y $w(e^*) < w(e)$ para toda arista $e \in E$, con $e \neq e^*$ (donde $w(e)$ denota el peso asignado a la arista e). Si aplicamos el algoritmo de Dijkstra a G y calculamos la distancia más corta de v_1 a cada uno de los vértices v_i para $2 \leqslant i \leqslant n$, entonces existe un vértice v_j, para algún $2 \leqslant j \leqslant n$, tal que la arista e^* se use en el camino más corto de v_1 a v_j.

b) El algoritmo de Dijkstra encuentra la longitud de una ruta más corta en un grafo conexo ponderado incluso si algunos pesos son negativos.

c) Dados por entrada un grafo conexo ponderado $G = (V, E, w)$ con $w(e) \geqslant 0$ para cada $e \in E$ y $a, z \in V$, el siguiente algoritmo regresa la longitud de una ruta más corta de a a z.

ESCUELA DE CIENCIAS EXACTAS Y NATURALES
DEPARTAMENTO DE MATEMÁTICA
Av. Pellegrini 250. Rosario
+54 0341 - 480 2649 internos 216 - 119

```
algo (w, a, z) {
    longitud = 0
    v = a
    T = conjunto de todos los vertices
    while (v not in z) {
        T = T - {v}
        seleccionar x in T con w(v,x) minimo
        longitud = longitud + w(v,x)
        v = x
    }
    return longitud
}
```